HASH 알고리즘

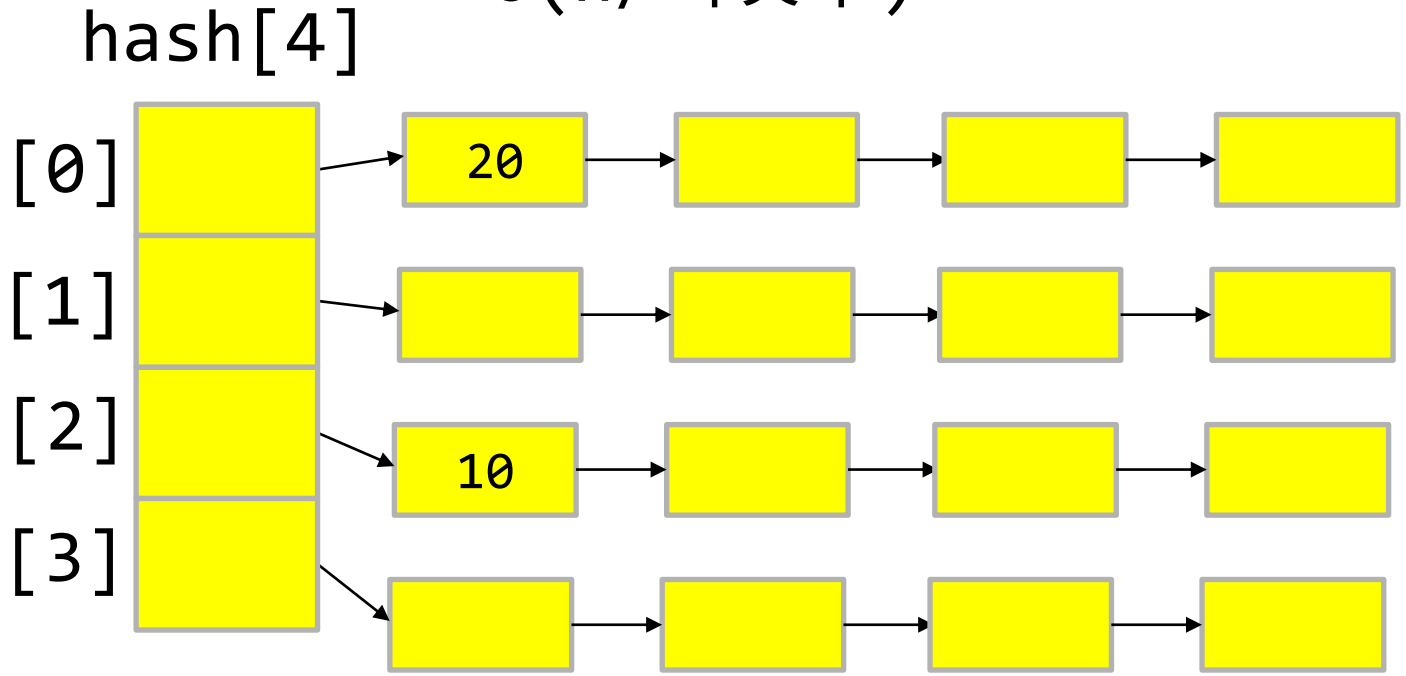
- **Hash** 알고리즘의 구동 원리를 이해한다.
- 최적화된 **hash** 함수의 구현원리를 이해한다.

1) Hash 알고리즘의 기본개념
2) Hash 함수의 최적화 원리 이해

hash 검색 알고리즘
: 분할된 linked list로 검색성능을 간단한 구현으로 향상 시킬 수 있다.

O(n/버킷수)

20 % 4 => 0



Generic Hash 타입 설계 및 구현

```
struct hlist_head {
    struct hlist_node *first;
};

struct hlist_node {
    struct hlist_node *next, **pprev;
};

typedef struct
{
    int sno;
    struct hlist_node hash;
} SAWON;
```

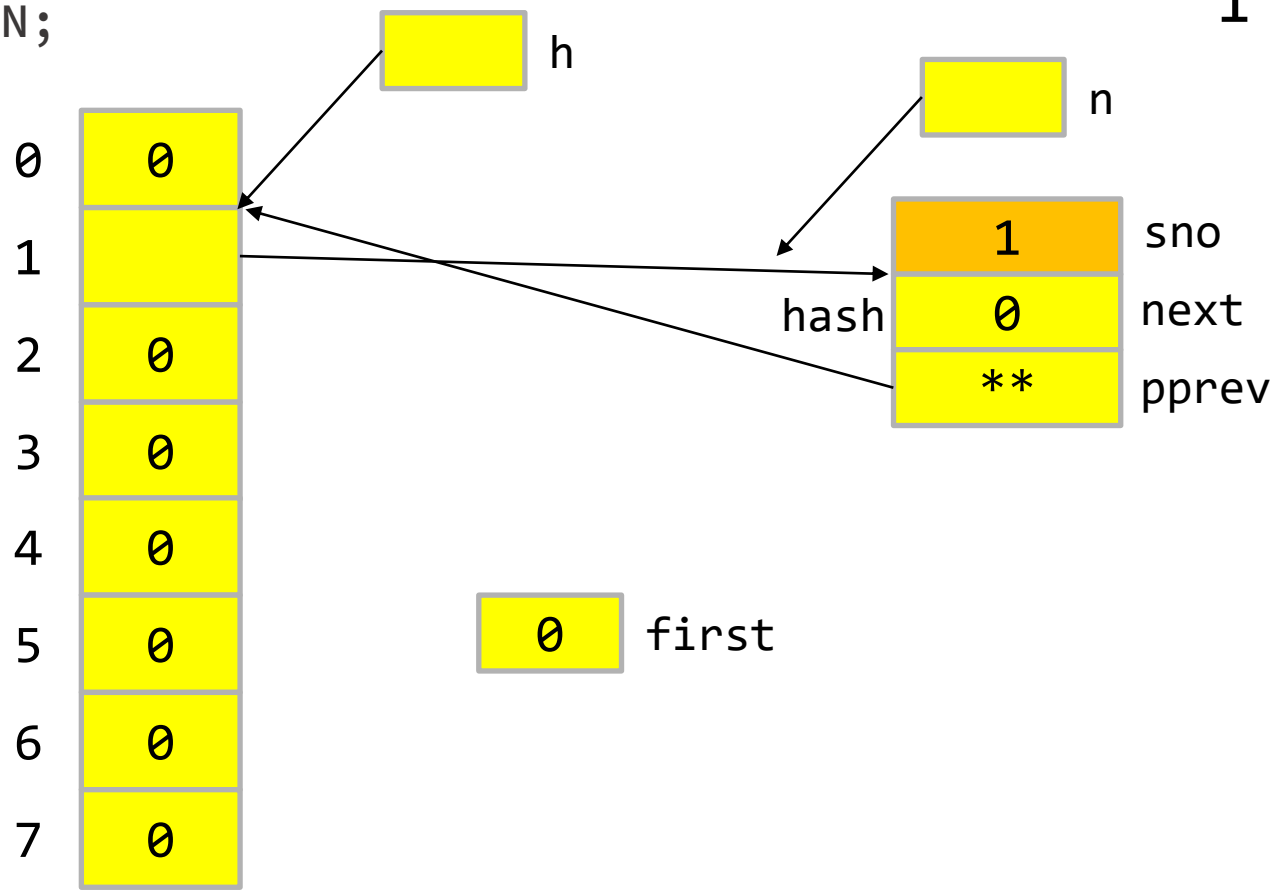
hash 검색 알고리즘 삽입 분석

```
void hlist_add_head(struct hlist_node *n,
                    struct hlist_head *h)
{
    struct hlist_node *first = h->first;
    n->next = first;
    if (first)
        first->pprev = &n->next;
    h->first = n;
    n->pprev = &h->first;
}
```

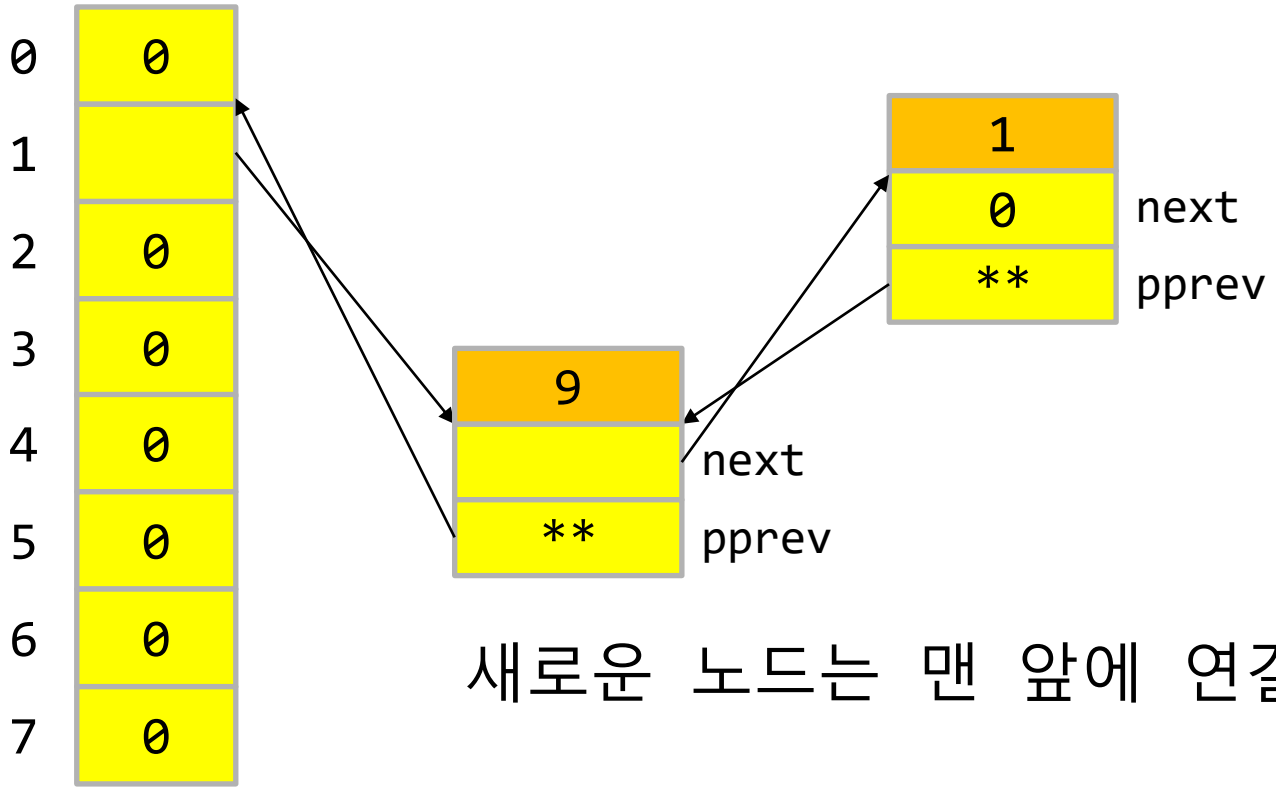
hash 검색 알고리즘 삽입 분석

```
typedef struct
{
    int sno;
    struct hlist_node hash;
} SAWON;
```

1 % 8 => 1



hash 검색 알고리즘 삽입 분석



새로운 노드는 맨 앞에 연결 된다.

hash 검색 알고리즘 삽입 분석

```
#define pid_hashfn(nr, ns)  \
    hash_long((unsigned long)nr + (unsigned long)ns,
                pidhash_shift)
#define hash_long(val, bits) hash_32(val, bits)

unsigned int pidhash_shift = 3;

pid_hashfn(sno);
hash_long(sno, 3);
hash_32(sno, 3)

// 랜덤수는 매우 큰 소수를 곱한후 최상위 비트를 추출 하면
// 더 최선의 수를 구할 수 있다.

u32 hash_32(u32 val=100, unsigned int bits=3)
{
    u32 hash = val * GOLDEN_RATIO_PRIME_32;
    return hash >> (32 - bits);
}
```